

Extensions

CGExtensions

frame 으로 UI입을 때

시점에 상관없이 View의 크기에 해당하는 값을 UIScreen 값으로 가져오기

```
extension CGRect {
    static var screenBounds: CGRect { return UIScreen.main.bounds }

    // 프레임으로 UI 잡을 때 make 만으로
    static func make(_ x: CGFloat, _ y: CGFloat, _ width: CGFloat, _ height: CGFloat) -> CGRect {
        return CGRect(x: x, y: y, width: width, height: height)
    }
}
```

size 가져오기

```
// 회전된 화면이 아닌 기본 세로 방향의 width 와 height 라는 것. 유의.
extension CGFloat {
    public static let screenWidth = UIScreen.main.nativeBounds.size.width / UIScreen.main.nativeScale
    public static let screenHeight = UIScreen.main.nativeBounds.size.height / UIScreen.main.nativeScale
}
```

UIViewExtensions

배열로 addsubview 하기

```
extension UIView {

    func addSubviews(_ views: [UIView]) {
        views.forEach { addSubview($0) }
    }
}
```

AutoLayoutAnchor 쉽게잡기.

frame 기반에서 UI 설정하기 쉽게 하기

```
extension UIView {
    var x: CGFloat {
        // self.x 로 쓰면되
        get { return frame.origin.x }
        set { frame.origin.x = newValue }
    }

    var y: CGFloat {
        get { return frame.origin.y }
        set { frame.origin.y = newValue }
    }

    var width: CGFloat {
        get { return frame.width }
        set { frame.size.width = newValue }
    }

    var height: CGFloat {
        get { return frame.height }
        set { frame.size.height = newValue }
    }

    var origin: CGPoint {
        get { return frame.origin }
        set { frame.origin = newValue }
    }

    var size: CGSize {
        get { return frame.size }
        set { frame.size = newValue }
    }

    var maxX: CGFloat {
        return frame.origin.x + frame.size.width
    }

    var maxY: CGFloat {
        return frame.origin.y + frame.size.height
    }
}
```

UIControlExtensions

addTarget 간단하게 쓰기

```
extension UIControl {
    func addTarget(action: Selector) {
        guard let vc = parentViewController else {
            fatalError("addSubview 메서드 호출 이후에 사용해야 함")
        }
        addTarget(vc, action: action, for: .touchUpInside)
    }

    // MARK: - 자기가 속해있는 View Controller 를 알 수 있다.
    var parentViewController: UIViewController? {
        var responder: UIResponder? = self
        while let nextResponder = responder?.next {
            responder = nextResponder
            // 건너 건너 vc 최 상위의 뷰 다음에 View Controller 가 나오면
            if let vc = nextResponder as? UIViewController {
                return vc
            }
        }
        return nil
    }
}
```

사용 전

```
button.addTarget(self, action: #selector(abs(_:)), for: .touchUpInside)
```

사용 후

```
button.addTarget(action: #selector(abs(_:)))
```

TableViewCollectionViewExtensions

regiser 해서 cell id 자동입력하기

사용 이전

```
collectionView.register(MenuCell.self, forCellWithReuseIdentifier: "menuCell")
```

사용 이후

```
collectionView.register(cell: MenuCell.self)
```

dequeue cell id 자동입력해주고, 타입캐스팅 자동으로 해줌

사용 이전

```
let cell = collectionView.dequeueReusableCell(
    withReuseIdentifier: "menuCell",
    for: indexPath) as! MenuCell
```

사용 이후

```
let cell = collectionView.dequeue(MenuCell.self, indexPath)
```

tableView 의 경우 indexPath 도 필요없이 Cell 의 타입만 넣어주면 끝

```
let cell = tableView.dequeue(CurrentForecastCell.self)
```